Remarks

In the specification, paragraphs on pages 6 and 10 have been amended to correct minor typographical problems.

Claims 1-16 were presented in Amendment A, and by the present office action, claims 1-16 were rejected. Claims 1-16 have been amended, and new claims 17-20 have been added. Thus, claims 1-20 remain in this application.

General Comments

The Examiner rejected claims 1-16 using various combinations of the following U.S. Patents: Dubey (Patent no. 5,724,565); Kimmel (Patent no. 6,105,053); and Tremblay (Patent no. 6,343,348). Applicant has reviewed each of these patents and believes that the remaining claims, as amended, are patentably distinct from these patents, taken alone or in combination. However, before beginning a claim by claim analysis of the remaining claims, and the examiner's rejections, applicant believes it beneficial to provide a brief overview of Dubey, and the present application.

Dubey discloses a multithreaded processor with the following arrangement (please refer to Figure 1 of Dubey). An instruction cache 110 has several ports 115 to allow simultaneous retrieval of multiple instructions for multiple instruction threads. The addresses used by the ports 115 for retrieving instructions are located in multiple program counters 120. The retrieved instructions are provided to multiple dispatchers 140, which act to update the multiple program counters 120. Thus, instruction ports 115 are coupled directly to program counters 120 and to dispatchers 140. Each of the dispatchers 140 are used to fetch and dispatch instructions for a particular program thread. The main program thread is handled by dispatcher 140-1. Future threads are controlled by dispatchers 140-2 thru 140-N. Instructions from each of the dispatchers 140 are provided to a scheduler 150. The scheduler 150 provides instructions received from the dispatchers 140 to multiple functional units 180, for execution. In one embodiment, the functional units all execute the same instructions from a common scheduler. In an alternative embodiment, the functional units execute different instructions (such as Integer instructions, Floating Point Instructions, etc.), where each functional unit group has its own scheduler 150. (see Col. 8, lines 42-45).

In contrast to Dubey, Applicant's invention discloses (see Figure 3 of the present application), an instruction cache 31 that has multiple ports, coupled to a single fetch unit 33. The fetch unit 33 is coupled to a plurality of queues 39. The queues 39 are coupled to a single dispatch stage 41, which in turn is coupled to multiple execution units 45. In this embodiment, several distinctions should be readily apparent between it and Dubey. First, there is no one-to-one correspondence between the number of threads executing, and the number of ports on the instruction cache 31, the number of fetch units 33, the number if dispatch stages 41, or the number of execution units 45. That is, unlike Dubey, although eight threads are shown to be executing in ten execution units 45, there is only one fetch unit 33, one dispatch unit 41, and two ports on the instruction cache 31. In Dubey's arrangement, he required a dispatch/fetch unit for every thread, and a port on his instruction cache for every thread. It is the *decoupling* of the fetching and the dispatching of multiple threads to which the present invention is directed. Nothing in Dubey allows such decoupling of fetching and dispatching, because he teaches a port per thread, a program counter per thread, and a fetch/dispatch unit per thread. Further, since Dubey does not appreciate the benefit of decoupling of the functions of fetching and dispatching, he does not appreciate the benefit providing by the instruction queues 39, which allow for the number of threads fetched in any cycle to differ from the number of threads dispatched. It is very important that the examiner understand this distinction, prior to analyzing the additional embodiment described with reference to Figure 4 of the present application. In Figure 4, applicant builds on the invention of Figure 3 (which allowed for decoupling of fetching from dispatch), to allow clustering of multiple threads into multiple clusters. Thus, one group of threads is clustered together into Cluster A 49, which shares two ports on the instruction cache 47, while a second group of threads is clustered together into Cluster B 51, which shares two ports on the instruction cache 47. In this arrangement, clustering of threads is provided for, while still decoupling the fetching of the threads from their dispatch. Moreover, as described in the specification, this decoupling is provided for in a clustered environment, whether the number of fetchers in each cluster is one, or one-to-one for each thread, and whether the number of dispatchers is one-to-one for each thread, or single for each cluster. Further, the clustered

9

threads are able to share a general execution unit 71, or utilize execution units specific for each cluster.

With these differences between Dubey and the present invention in mind, Applicant will now provide an analysis of the existing claims, with respect to the examiner's rejections.

Claim Rejections - 35 USC §103(a)

The examiner rejected claims 1-3, 5, 6, 8, 16 under 35 USC §103(a) as being unpatentable over Dubey in view of Kimmel.

With respect to claim 1, it is repeated below, as amended, for each of reference:

1. (Currently amended) A pipelined multistreaming processor, comprising:

an instruction cache having a plurality of ports:

a first cluster of a plurality of instruction streams, said first cluster fetching instructions from said instruction cache;

a second cluster of a plurality of instruction streams, said second cluster fetching instructions from said instruction cache;

a plurality of instruction queues, one for each of said instruction streams in each of said first and second clusters;

a first dispatch stage, coupled to said first cluster, for dispatching to execution units instructions from said instruction streams in said first cluster; and

a second dispatch stage, coupled to said second cluster, for dispatching to execution units instructions from said instruction streams in said second cluster;

wherein said first and second clusters operate independently, with said first and second dispatch stages taking instructions only from said plurality of instruction queues which are in said clusters to which said dispatch stages are dedicated; and

wherein said first and second clusters are coupled to said plurality of ports on said instruction cache, and can each utilize one or more of said plurality of ports.

The examiner indicated that Dubey teaches an instruction source. Applicant demurs.

The examiner further indicated that Dubey teaches a first cluster of a plurality of streams from the instruction source. Applicant respectfully traverses. As mentioned above, nothing in Dubey is directed at clustering of threads. Dubey can only be read to show clustering if ALL instruction threads are part of a first cluster OR, if each thread is its own cluster. Neither of these are a cluster. Dubey only shows a plurality of streams, each which are individual threads. Since Dubey does not address clustering of threads (plural), applicant respectfully suggests that Dubey does not show a first cluster of a plurality of instruction streams.

The examiner further states that Dubey teaches a second cluster of a plurality of streams. The examiner references Figures 1A and 1B, and col. 6, line 38 thru Col. 7, line 55 and Col. 8, lines 16-60. Applicant has examined these sections, as well as the rest of Dubey, and respectfully traverses. Teaching fetching and dispatch of multiple threads is not the same as teaching the clustering of multiple threads. See applicant's figures 3 and 4. Multiple threads are clustered together in applicant's invention. Nothing in Dubey shows or teaches clustering of threads. Splitting of schedulers per subsets of functional units is not the same as clustering of threads. Clustering of threads implies that all resources associated with a thread, from fetching, to dispatch, is part of the cluster. Nothing in Dubey shows this.

The examiner further states that Dubey teaches first and second dispatch stages for dispatching instructions to execution units. Applicant respectfully traverses. As mentioned above, Dubey shows a dispatch stage for each thread, not a dispatch stage coupled to a cluster. Nowhere does Dubey teach, suggest, or even hint at a dispatch stage coupled to a cluster, for dispatching to execution units instructions from said instruction streams (plural) in a cluster.

The examiner further indicates that each dispatcher in Dubey operates independently of the other dispatchers, and that in one embodiment, the schedulers were split to only

11

schedule the instructions from the corresponding dedicated dispatcher. Applicant agrees. Dubey does teach a common scheduler for all dispatchers, or splitting a scheduler into a set of schedulers, per functional block (see Col. 8, lines 43-45). Applicant notes that what is intended here is that if one set of functional blocks executes one instruction set (e.g., for floating point instructions), and another set of functional blocks executes another instruction set (e.g., for integer instructions), that he would have a scheduler for each functional block. What this does not mean, however, is that fetching, and dispatch of instruction threads are clustered. They are still handled individually, in Dubey, with their own dispatch unit. They are then passed to the schedulers, that make sure that they do not schedule a floating point instruction to be executed by an integer unit, or vice versa. This is NOT clustering of threads.

The examiner further indicated that Dubey did not detail that instruction buffers were instruction queues. Applicant agrees. The examiner further stated that Kimmel taught an individual queue and dispatcher for each execution unit. Applicant respectfully traverses. Kimmel is talking about an operating system software environment. To recite Kimmel (Col. 5, lines 43-45) "For each JP 100-107, the *operating system* establishes a run queue and a dispatcher [Note, these are software processes]. The dispatcher is a *kernel subsystem* that is a mechanism responsible for scheduling and executing processes on an associated JP ..." A run queue in an operating system is NOT the same as a physical instruction queue for holding instructions, much less an instruction queue which is coupled to anything physical, like a fetcher, or a dispatcher, even much less than an instruction queues that hold instruction streams that are clustered. Further, the dispatcher in Kimmel is NOT a physical dispatcher associated with an instruction stream, but rather, another operating system subsystem for dispatching processes. Thus, nothing in Kimmel is directed at the absence in Dubey related to instruction queues coupled to dispatchers. And, even if Kimmel did supplement Dubey with such teaching, there is nothing in either Dubey or Kimmel to suggest the combination. Kimmel is not directed at a multithreaded processor, and Dubey is not directed operating system software.

The examiner states that it "would have been obvious to one of ordinary skill in the DP art to combine the teachings of Dubey and Kimmel. Applicant respectfully traverses. Applicant respectfully submits that the art of Kimmel (operating systems), and the art of

12

Dubey (processors) is different. Applicant further submits that the patent office agrees with this position, because the class/subclass for Kimmel is 709/105, and the class/subclass for Dubey is 395/595. But, even if they were classified the same, that still would not be enough for the combination. As the Federal Circuit has held, "The mere fact that the prior art may be modified in the manner suggested by the Examiner does not make the modification obvious unless the prior art *suggested the desirability* of the modification.... It is impermissible to use the claimed invention as an instruction manual or "template" to piece together the teachings of the prior art so that the claimed invention is rendered obvious.... One cannot use hindsight reconstruction to pick and choose among isolated disclosures in the prior art to deprecate the claimed invention." In re Fritch, 872 F.2d 1260, 23 USPQ2d 1780 (Fed. Cir. 1992) at 1783-1784. Thus, applicant respectfully suggests that even if Kimmel taught the instruction queues / dispatch arrangement as claimed, and as missing from Dubey, that it could not be combined in hindsight with Dubey because there is no suggestion in either Dubey or Kimmel for the combination.

For all of these reasons, applicant respectfully requests the examiner to withdraw his rejection of claim 1.

With respect to claim 2-8, these depend from claim 1 and add further limitations that are neither anticipated nor obviated by Dubey, taken alone or in combination with Kimmel. For all of the reasons above, applicant respectfully requests the examiner to withdraw his rejection of these claims.

The examiner utilizes Tremblay, in combination with Dubey and Kimmel to reject claims 4,7 under 35 USC §103(a). Applicant respectfully traverses. The examiner states that Tremblay taught "a system with eight streams executed on eight execution units and four streams per cluster". Applicant respectfully suggests that the examiner is incorrect. What Tremblay shows/teaches are two processors, that can, in parallel, process one thread each, where each tread consists of four instructions. Tremblay states "Two media processing units 110 and 112 are included in a single integrated circuit chip to support an execution environment exploiting thread level parallelism in which *two independent threads can execute simultaneously.*" Col. 5, lines 24-27. Tremblay recognizes that it is

desirable to execute multiple threads in parallel, and provides two processors, each of which execute a single thread, but which together execute two threads in parallel. Nothing in Tremblay is directed at clustering of threads, or at parallel execution of multiple clusters. For these reasons, and for those stated above with respect to claim 1, applicant respectfully requests the examiner to withdraw his rejections of claims 4 and 7.

The examiner rejected claims 9-11, 13 and 15 under 35 USC §103(a) as being unpatentable over Dubey in view of Kimmel. Applicant respectfully traverses. Applicant's comments above regarding the teaching of Dubey and Kimmel need not be repeated. However, it is considered relevant to repeat claim 9, as amended, for ease of reference:

9. (Currently amended) In a pipelined multistreaming processor having an instruction cache which has a plurality of ports, and a plurality of instruction streams executing within the processor, a method for clustering ones of the plurality of instruction streams, comprising:

clustering a first plurality of the instruction streams into a first cluster; clustering a second plurality of the instruction streams into a second cluster, the first and second plurality of the instruction streams being independent;

providing a first dispatch stage to the first plurality of instruction streams of the first cluster, for dispatching the first plurality of instructions to first execution units;

providing a second dispatch stage to the second plurality of instruction streams of the second cluster, for dispatching the second plurality of instructions to second execution units; and

in each cycle, fetching, instructions from the instruction cache for one of the instruction streams in each of the first and second cluster.

Claim 9 specifically recites clustering a first plurality of threads into a first cluster, clustering a second plurality of threads into a second cluster, providing dispatch stages for each cluster, and fetching instructions for streams in each cluster, in a cycle. As

14

mentioned above, nothing in Dubey, taken alone or in combination with Kimmel, teaches clustering of threads, much less, fetching of instructions from threads within each of the clusters in a given cycle. For all of the reasons stated above with respect to claim 1, and for these reasons, applicant respectfully requests the examiner to withdraw his rejection of this claim.

With respect to claims 11-16, these depend from claim 9 and add further limitations that are neither anticipated nor obviated by Dubey, taken alone or in combination with Kimmel. For all of the reasons stated above, applicant respectfully requests the examiner to withdraw his rejection of these claims.

With respect to claims 12, 14, the examiner further rejected these claims under 35 USC §103(a) as being unpatentable over Dubey in view of Kimmel and further in view of Tremblay. Applicant respectfully traverses for the reasons stated above with respect to claims 4 and 7. Applicant therefore respectfully requests the examiner to withdraw his rejection of claims 12 and 14.
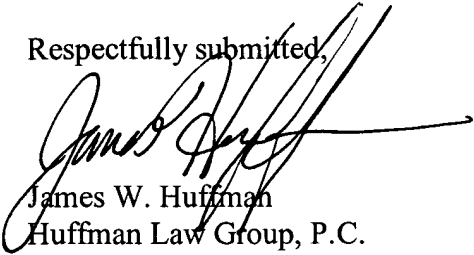
Applicant has added new claims 17-20 which applicant believes are novel and non-obvious in view of Dubey, Kimmel and Tremblay, taken alone or in combination.

Applicant respectfully requests that the Examiner withdraw his rejection of claims 1-16, and allow these claims, as well as the newly added claims 17-20.

Applicant earnestly requests the examiner to telephone him at the direct dial number printed below if the examiner has any questions or suggestions concerning the application.

Respectfully submitted,

[signature]

James W. Huffman
Huffman Law Group, P.C.
Registration No. 35,549
Customer No. 23669
1832 N. Cascade Ave.
Colorado Springs, CO 80907
719.475.7103
719.623.0141 fax
jim@huffmanlaw.net

Date: 9-22-04

"EXPRESS MAIL" mailing label number ED34765 0055US. Date of Deposit 9-22-04. I hereby certify that this paper is being deposited with the U.S. Postal Service Express Mail Post Office to Addressee Service under 37 C.F.R. §1.10 on the date shown above and is addressed to the U.S. Commissioner of Patents and Trademarks, Alexandria, VA, 22313.

By: *[signature]*

16